## AMENDMENTS TO THE SPECIFICATION

Please replace paragraphs [0006], [0012], [0031], and [0063] of the specification with the following replacement paragraphs:

**[0006]** Various remote applications or systems often attempt to update and/or acquire PLC information or related device information *via* a plurality of different, competing and often incompatible or insecure network technologies. A major concern with this type of access to PLC[[']]s and control systems in general, relates to the amount of security that is provided when sending or receiving data to and from the PLC and/or associated equipment. In most factories or industrial environments, complex and sometimes dangerous operations are performed in a given manufacturing setting. Thus, if a network-connected controller were inadvertently accessed, or even worse, intentional sabotage were to occur by a rogue machine or individual, potentially harmful results can occur.

**[0012]** One function of the network validation tools is to perform vulnerability scanning and auditing on the networks. This includes checking for susceptibility to common network-based attacks, searching for open TCP/UDP ports, and scanning for vulnerable network services. The tools can also attempt to gain key identity information about end devices that may enable hacker entry. Another function of the network validation tools is to perform vulnerability scanning and auditing on firewalls, routers, and/or other network/security devices. In addition, a complementary tool can be provided to assess CIP-based factory automation systems for security. This will typically be a network-based tool, since factory automation devices often are not as capable as general purpose computing devices. The tool can also be operable in an assessment mode to discover system flaws with little or no configuration, and the tool can operate in a validation mode to check system security against security analysis methodology determinations described above. Still [[yet]] other functions can include non-destructively mapping a topology of IT and automation devices, checking revisions and configurations, checking user attributes, and/or checking access control lists. The validation tools described herein can also be adapted to automatically correct security problems (*e.g.,* automatically adjust security parameters/rules/policies, install new security components, remove suspicious components, and so forth).

**[0031]** Referring initially to Fig. 1, a system 100 illustrates various automation security tools in accordance with an aspect of the present invention. One or more automation assets 120 communicate and cooperate with various network devices 124 across a network 130. The automation assets 120 include substantially any type of control, communications module, computer, I/O device, Human Machine Interface (HMI)[[)]] that communicate *via* the network 130 which includes control, automation, and/or public networks. In one example, the automation assets 120 include Programmable Logic Controllers (PLCs) that can also communicate to and control various other assets such as Input/Output modules including Analog, Digital, Programmed/Intelligent I/O modules, other programmable controllers, communications modules, and the like. The network 130 includes public networks such as the Internet, Intranets, and automation networks such as Control and Information Protocol (CIP) networks including DeviceNet and ControlNet. Other networks 130 include Ethernet, DH/DH+, Remote I/O, Fieldbus, Modbus, Profibus, wireless networks, serial protocols, and so forth. In addition to the automation assets 120, the network devices 124 include various possibilities (hardware and/or software components). These include components such as switches with virtual local area network (VLAN) capability, LANs, WANs, proxies, gateways, routers, firewalls, virtual private network (VPN) devices, intrusion detection systems, servers, clients, computers, configuration tools, monitoring tools, and/or other devices.

**[0063]** In order to process the training data 810, the learning component 800 includes one or more learning models 820 and/or learning variables 830. As noted above, the learning models 820 can include such aspects as neural network functions, inference models, mathematical models, statistical models, probabilistic models, classifiers, and so forth that learn network patterns or occurrences from the training data 810. It is also noted that the learning models can be adapted similarly (*e.g.,* all models configured as Hidden Markov Models) or adapted in various combinations (*e.g.,* 40 models configured as a neural network, [[3]] three models adapted in a Bayesian configuration, [[1]] one model configured as a vector-based classifier). The learning variables 830 can be focused on selected events or circumstances. For example, a network load variable may record the average number of outside network requests per hour. In another example, a PLC variable may record the average number of network retries that an associated PLC experiences in a given timeframe, whereas another PLC variable records the maximum number of network retries that the PLC experienced during the same timeframe. In another aspect, the learning variables 820 may be employed as counters to record amounts for various events (*e.g.,* record the number of PLC network transfers to I/O device over the last hour). As can be appreciated, a plurality of such variables can be defined and updated to log various network events during a selected training period. After training, the learning component 810 stores learned patterns or events that are then employed by a learning analyzer component described below to monitor and detect network security problems or identify potential security issues.